|  |  | Committed Backlog Items | Tasks | Who? | Not Started | In Progress | Completed |
|---|---|---|---|---|---|---|---|
| **PBI #1** |  | **PBI #1 - Generate error messages** | **PBI time: 8hr** | **Alex** |  | X |  |
| *As* | an election official using the voting system software | 0.5hr | Catch error conditions | Alex |  |  | X |
| *I want* | the program to provide appropriate error messages if a bad ballot file is passed in | 0.5hr | Generate useful error messages | Alex |  |  | X |
| *So that* | I can fix the file and provide the correctly formatted file | 0.5hr | Cleanly resolve error | Alex |  |  | X |
|  |  | 4hr | Fuzz via AFL++ | Alex |  |  | X |
| *Acceptance Criteria* | Generate appropriate errors for empty ballot files, files with parse problems | 0.5hr | Documentation | Alex |  |  | X |
| *Definition of Done* | Generate an error if unable to parse input CSV file; accept correctly formatted CSV files; No compilation error; unit tests for each method and agreed amount of system tests pass; correct output | 1hr | Write new unit tests | Alex |  |  | X |
| *Effort* | Medium (~5-6 hours) | 0.1hr | Run new unit tests | Alex |  |  | X |
|  |  | 0.5hr | Write new system tests | Alex |  |  | X |
|  |  | 0.1hr | Run new system tests | Alex |  |  | X |
|  |  | 0.15hr | Write unit test logs | Alex | X |  |  |
|  |  | 0.15hr | Write system test logs | Alex | X |  |  |
| **PBI #2** |  | **PBI #2 - Resolve ties via coin flip** | **PBI time: 3.5hr** | **Alex, Janani** |  | X |  |
| *As* | an election official using the voting system software | 1hr | Add new functionality | Janani |  |  | X |
| *I want* | Candidates or parties with ties to be resolved via a 50/50 coin flip | 0.5hr | Documentation | Janani |  |  | X |
| *So that* | Ties ae resolved randomly, fairly, and indepdently of other factors | 1hr | Write new unit tests | Alex | X |  |  |
|  |  | 0.1hr | Run new unit tests | Alex | X |  |  |
| *Acceptance Criteria* | Ties between candidates or between parties can be resolved randomly | 0.5hr | Write new system tests | Alex | X |  |  |
| *Definition of Done* | Ties between candidates or between parties can be resolved randomly; No compilation error; unit tests for each method and agreed amount of system tests pass; correct output | 0.1hr | Run new system tests | Alex | X |  |  |
| *Effort* | Small (~2-3 hours) | 0.15hr | Write unit test logs | Janani | X |  |  |
|  |  | 0.15hr | Write system test logs | Janani | X |  |  |
| **PBI #3** |  | **PBI #3 - VotingSystem api has one method for elections** | **PBI time: 2.6hr** | **Alex** |  |  | X |
| *As* | a developer | 0.1hr | Rewrite old calls in Election | Alex |  |  | X |
| *I want* | to be able to create and run an election, easily. The electables field is not populated unless three additional functions are celled in order | 0.5hr | Documentation | Alex |  |  | X |
| *So that* | The probability of logic errors are reduced and the system is easily extensible. | 1hr | Write new unit tests | Alex |  |  | X |
|  |  | 0.1hr | Run new unit tests | Alex |  |  | X |
| *Acceptance Criteria* | A new method is added to votingSystem to automatically call processElectables(), countVotes(), and calculateResults(). | 0.5hr | Write new system tests | Alex |  |  | X |
| *Definition of Done* | No compilation error; unit tests for each method and agreed amount of system tests pass; correct output | 0.1hr | Run new system tests | Alex |  |  | X |
| *Effort* | Small (~2-3 hours) | 0.15hr | Write unit test logs | Alex |  |  | X |
|  |  | 0.15hr | Write system test logs | Alex |  |  | X |
| **PBI #4** |  | **PBI #4 - Multiple files for the CPL algorithm** | **PBI time: 3.5hr** | **Ashwin, Alex, Leo** |  | X |  |
| *As* | an election official | 0.25hr | Change CLI functionality to take in multiple files for CPL | Alex |  |  | X |
| *I want* | Multiple files to be brought in for a CPL election | 0.25hr | Change GUI functionality to take in multiple files for CPL | Alex |  |  | X |
| *So that* | they can bring in different files from different balloting locations | 1hr | Code functionality for multiple files read | Leo |  |  | X |
|  |  | 0.5hr | Write new unit tests for new methods for CPL multiple file functionality | Ashwin | X |  |  |
| *Acceptance Criteria* | Multiple .csv input files for CPL election being read correctly at the same time | 0.1hr | Run new unit tests for new methods written | Ashwin | X |  |  |
| *Definition of Done* | No compilation error; unit tests for each method and agreed amount of system tests pass; correct output | 0.5hr | Write integration tests for CPL multiple files | Leo |  |  | X |
| *Effort* | Small (~2-3 hours) | 0.1hr | Run Integration tests to make sure system still functions correctly | Leo |  |  | X |
|  |  | 0.3hr | Write test logs for unit and integration tests | Leo |  |  | X |
|  |  | 0.5hr | Write Doxygen documentation for new methods written | Ashwin | X |  |  |
| **PBI #5** |  | **PBI #5 - Multiple files for the OPL algorithm** | **PBI time: 3.5hr** | **Ashwin, Leo** |  | X |  |
| *As* | an election official | 0.25hr | Change CLI functionality to take in multiple files for OPL | Ashwin |  |  | X |
| *I want* | Multiple files to be brought in for an OPL election | 0.25hr | Change GUI functionality to take in multiple files for OPL | Ashwin |  |  | X |
| *So that* | they can bring in different files from different balloting locations | 1hr | Code functionality for multiple files read | Leo |  |  | X |
|  |  | 0.5hr | Write new unit tests for new methods for OPL multiple file functionality | Leo |  |  | X |
| *Acceptance Criteria* | Multiple .csv input files for OPL election being read correctly at the same time | 0.1hr | Run new unit tests for new methods written | Leo |  |  | X |
| *Definition of Done* | No compilation error; unit tests for each method and agreed amount of system tests pass; correct output | 0.5hr | Write integration tests for OPL multiple files | Leo |  |  | X |
| *Effort* | Small (~2-3 hours) | 0.1hr | Run Integration tests to make sure system still functions correctly | Leo |  |  | X |
|  |  | 0.3hr | Write test logs for unit and integration tests | Ashwin | X |  |  |
|  |  | 0.5hr | Write Doxygen documentation for new methods written | Leo |  |  | X |
| **PBI #6** |  | **PBI #6 - MPO files in general** | **PBI time: 3hr** | **Janani, Leo** |  |  |  |
| *As* | an election official | 0.5hr | MPO can read MPO file | Janani |  |  | X |
| *I want* | MPO election ballots to be brought in via a .csv file | 0.5hr | MPO correctly inherits from VotingSystem and implements all required methods | Janani |  |  | X |
| *So that* | the election can be run by the MPO algorithm in the system | 0.5hr | Write unit tests for added functionality | Janani |  |  | X |
|  |  | 0.1hr | Run unit tests | Janani |  |  | X |
| *Acceptance Criteria* | .csv input file for MPO election being read correctly | 0.5hr | Write integration tests to make sure the system still works as intended | Leo |  |  | X |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| *Definition of Done* | No compilation error; unit tests for each method and agreed amount of system tests pass; correct output | 0.1hr | Run integration tests | Leo | | | X |
| *Effort* | Small (~2-3 hours) | 0.15hr | Write unit test logs | Janani | | X | |
| | | 0.15hr | Write system test logs | Janani | | X | |
| | | 0.5hr | Write Doxygen documentation for new methods written | Janani | | X | |
| **PBI #7** | | **PBI #7 - MPO multiple files** | **PBI time: 2.7hr** | **Alex, Ashwin, Leo** | | X | |
| *As* | an election official | 0.1hr | Change CLI functionality to take in multiple files for MPO | Alex | | | X |
| *I want* | Multiple files to be brought in for a MPO election | 0.1hr | Change GUI functionality to take in multiple files for MPO | Alex | | | X |
| *So that* | they can bring in different files from different balloting locations | 0.5hr | Code functionality for multiple files read | Leo | | | X |
| | | 0.5hr | Write new unit tests for new methods for MPO multiple file functionality | Ashwin | | | X |
| *Acceptance Criteria* | Multiple .csv input files for MPO election being read correctly at the same time | 0.1hr | Run new unit tests for new methods written | Ashwin | X | | |
| *Definition of Done* | No compilation error; unit tests for each method and agreed amount of system tests pass; correct output | 0.5hr | Write integration tests for MPO multiple files | Ashwin | | | X |
| *Effort* | Small (~2-3 hours) | 0.1hr | Run Integration tests to make sure system still functions correctly | Ashwin | X | | |
| | | 0.3hr | Write test logs for unit and integration tests | Leo | | X | |
| | | 0.5hr | Write Doxygen documentation for new methods written | Leo | | X | |
| **PBI #8** | | **PBI #8 - MPO election** | **PBI time: 4.5hr** | **Alex, Leo, Ashwin** | | X | |
| *As* | an election official | 1hr | Break RawData::ParseFile() into sub functions | Alex | | | X |
| *I want* | to write an algorithm that determines multiple winners based on the candidates receiving the most ballots with a fair coin toss | 0.5hr | Change header line read order based on election type | Alex | | | X |
| *So that* | an MPO election can be run by the voting system | 1hr | Add new functionality | Leo, Janani | | | X |
| | | 0.5hr | Write unit tests for added functionality | Leo | | | X |
| *Acceptance Criteria* | MPO algorithm correctly calculates the winners and losers and fairly determines ties if there are any | 0.1hr | Run unit tests | Leo | | | X |
| *Definition of Done* | No compilation error; unit tests for each method and agreed amount of system tests pass; correct and fair output | 0.5hr | Write integration tests to make sure the system still works as intended | Leo | | | X |
| *Effort* | Large (~6-7 hours) | 0.1hr | Run integration tests | Leo | | | X |
| | | 0.15hr | Write unit test logs | Ashwin | | X | |
| | | 0.15hr | Write system test logs | Ashwin | | X | |
| | | 0.5hr | Write Doxygen documentation for new methods written | Ashwin | | X | |
| **PBI #9** | | **PBI #9 - MV files in general** | **PBI time: 3hr** | **Ashwin, Janani** | | | |
| *As* | an election official | 0.5hr | RawData can read MV file | Ashwin | | X | |
| *I want* | MV election ballots to be brought in via a .csv file | 0.5hr | Fileops can read MV file | Ashwin | | X | |
| *So that* | the election can be run by the MV algorithm in the system | 0.5hr | Write unit tests for added functionality | Ashwin | | X | |
| | | 0.1hr | Run unit tests | Ashwin | | X | |
| *Acceptance Criteria* | .csv input file for MV election being read correctly | 0.5hr | Write integration tests to make sure the system still works as intended | Janani | | X | |
| *Definition of Done* | No compilation error; unit tests for each method and agreed amount of system tests pass; correct output | 0.1hr | Run integration tests | Janani | | X | |
| *Effort* | Small (~2-3 hours) | 0.15hr | Write unit test logs | Janani | | X | |
| | | 0.15hr | Write system test logs | Janani | | X | |
| | | 0.5hr | Write Doxygen documentation for new methods written | Ashwin | | X | |
| **PBI #10** | | **PBI #10 - MV multiple files** | **PBI time: 2.7hr** | **Alex, Leo** | | X | |
| *As* | an election official | 0.1hr | Change CLI functionality to take in multiple files for MV | Alex | | | X |
| *I want* | Multiple files to be brought in for an MV election | 0.1hr | Change GUI functionality to take in multiple files for MV | Alex | | | X |
| *So that* | Different files can be brought in from different balloting locations | 0.5hr | Code functionality for multiple files read | Leo | | | X |
| | | 0.5hr | Write new unit tests for new methods for MV multiple file functionality | Leo | | | X |
| *Acceptance Criteria* | Multiple .csv input files for MV election being read correctly at the same time | 0.1hr | Run new unit tests for new methods written | Leo | | | X |
| *Definition of Done* | No compilation error; unit tests for each method and agreed amount of system tests pass; correct output | 0.5hr | Write integration tests for MV multiple files | Leo | | | X |
| *Effort* | Small (~2-3 hours) | 0.1hr | Run Integration tests to make sure system still functions correctly | Leo | | | X |
| | | 0.3hr | Write test logs for unit and integration tests | Leo | | | X |
| | | 0.5hr | Write Doxygen documentation for new methods written | Leo | | | X |
| **PBI #11** | | **PBI #11 - MV election** | **PBI time: 4.5hr** | **Alex, Janani, Ashwin** | | X | |
| *As* | an election official | 1hr | Break RawData::ParseFile() into sub functions | Alex | | | X |
| *I want* | to determine multiple winners based on the candidates receiving the most ballots with a fair coin toss if there is a tie or ties between candidate | 0.5hr | Change header line read order based on election type | Alex | | | X |
| *So that* | our voting system can handle and process votes case in an MV election using an Municipality Voting (MV) algorithm | 1hr | Add new functionality | Janani, Ashwin | | X | |
| | | 0.5hr | Write unit tests for added functionality | Ashwin | | X | |
| *Acceptance Criteria* | Program correctly produces multiple winners in MV format | 0.1hr | Run unit tests | Ashwin | X | | |
| *Definition of Done* | Program section passes unit tests; no logic errors that produces incorrect winner results; agreed amount of system tests pass | 0.5hr | Write integration tests to make sure the system still works as intended | | X | | |
| *Effort* | Large (~6-7 hours) | 0.1hr | Run integration tests | | X | | |
| | | 0.15hr | Write unit test logs | Ashwin | X | | |
| | | 0.15hr | Write system test logs | Ashwin | X | | |
| | | 0.5hr | Write Doxygen documentation for new methods written | Ashwin | X | | |
| **PBI #12** | | **PBI #12 - MPO Display** | **PBI time: 3hr** | **Alex,** | | X | |
| *As* | an election official | 1hr | Add new functionality | Alex | | | X |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| *I want* | MPO election statistics to be displayed to the screen at the end of the election | 0.5hr | Write unit tests for added functionality (Move from election) | Alex | | X | |
| *So that* | MPO election results and other relevant info can be read and understood easily by election officials | 0.1hr | Run unit tests | | X | | |
| | | 0.5hr | Write integration tests to make sure the system still works as intended | | X | | |
| *Acceptance Criteria* | Clearly displays the winners and losers of the election, as well as the percentage of votes each candidate received. | 0.1hr | Run integration tests | | X | | |
| *Definition of Done* | No compilation error; unit tests for each method and agreed amount of system tests pass; correct output/displays the correct information in an easily understandable format | 0.15hr | Write unit test logs | | X | | |
| *Effort* | Medium (~4-5 hours) | 0.15hr | Write system test logs | | X | | |
| | | 0.5hr | Write Doxygen documentation for new methods written | | X | | |
| **PBI #13** | | **PBI #13 - MV Display** | **PBI time: 3hr** | **Alex,** | | X | |
| *As* | an election official | 1hr | Add new functionality | Alex | | | X |
| *I want* | to display results from an MV election to the screen to see who won and who lost | 0.5hr | Write unit tests for added functionality (Move from election) | Alex | | X | |
| *So that* | election officials can easily view results and know the number of votes they received by all the voters | 0.1hr | Run unit tests | | X | | |
| | | 0.5hr | Write integration tests to make sure the system still works as intended | | X | | |
| *Acceptance Criteria* | Displays the winners and losers of the election, as well as the number of votes received by all of the voters | 0.1hr | Run integration tests | | X | | |
| *Definition of Done* | No compilation error; unit tests for each method and agreed amount of system tests pass; correct output/displays the correct information in an easily understandable format | 0.15hr | Write unit test logs | | X | | |
| *Effort* | Medium (~4-5 hours) | 0.15hr | Write system test logs | | X | | |
| | | 0.5hr | Write Doxygen documentation for new methods written | | X | | |